

A Tcl/Tk Based Graphical Interface to Medical and Administrative Information

Charles Webster¹ MD, MSIE, MSIS, Andrew Pople², Roseanne Silva³, MSIN,
Xiaofeng Wang⁴, MS, Sean McLinden¹, MD

¹Dept. of Health Information Sciences, Duquesne University

²Carnegie Group Inc., ³Fiserv Inc., ⁴Shadyside Hospital, Pittsburgh PA

FELIX is an front-end application processor, with an open systems back-end, that provides a uniform and intuitive interface to clinical and administrative information. It consists of an information browser, three clinical applications, and three management applications. FELIX was developed in a community hospital environment, but has conceptual and technical roots in medical informatics and the Internet.

INTRODUCTION

FELIX (FELIX Enables Limitless Information Exploration) provides a generic graphical interface for browsing medical and administrative information. It relies on a large number of industry standards such as TCP/IP, X11R5, SQL, Postscript and SGML. In particular, the interface was developed using a high-level X-windows graphical scripting language called Tcl/Tk [1] (which is currently also being ported to Windows and Macintosh platforms). We used real patient information and based our targeted applications on analysis of patient care processes at a local urban community referral hospital. We intended FELIX to be a vision of what is possible, a prototype to force us to confront the necessary integration of disparate technologies, and an inducement for clinicians and administrators to press for open systems file formats, programmatic interfaces and network protocols.

FELIX applications, are manipulatable from within a common, but customizable, information browser. The three clinical applications are a clinical document viewer that uses SGML formatted files to generate Tcl/Tk scripts which are then executed, a digital image and audio transcription system (used in radiographic scenarios), and a hypertext facility for viewing pathology images while linking phrases in text to objects in images. The three administrative applications are a real-time resource monitor that is dynamically updated from bar code readers on a network (used in this case to represent the amount of time that patients, tracked by "wandering" them into rooms, have spent in their current locations), a quality control and trend analyzer that allows retrospective display of activities such as weekly radiology procedures by ordering physician, and a visually intuitive resource scheduler that intelligently represents possible and impossible resource combinations (physician, nurse, room, and equipment).

FELIX would not have been possible without Tcl (tool command language), an application independent, embeddable interpreted command language, and TK, an X11 tool kit and widgets based on Tcl. The combination provides a high-level graphical scripting language that can be easily extended by writing C programs that become new Tcl commands. Applications created with Tcl/Tk can allow a user to dynamically modify their interface while working from within it. Many of the capabilities that support FELIX functions are extensions to Tcl created either by us (in the case of digital audio) or by members of the Tcl/Tk community and made available over the Internet (in the case of the photo and hypertext widgets and extensions to provide additional access to the UNIX environment).

Some of the ideas we explored included a double layer of menus that was retrieved from a database and configured by the user in order to obtain customized views of database content. Each retrieved "document" is actually an executable Tcl/Tk script that is generated from an SGML file or from the contents of a database. This allowed us to embed useful behavior while retaining flexibility, such as embedding buttons that turn into digital signatures or text widgets that adjust their size to the number of lines contained.

While FELIX relied on industry standard formats, application programming interfaces, and network protocols, none of the potential data sources were accessible, except by manually down-loading and translating data from proprietary hospital and departmental information systems. However, FELIX was built to demonstrate to hospital clinicians and administrators what they could have on their desktops if they invested in the required network infrastructure, and made sure that future information systems, and modifications of extant information systems, complied with open system standards. As a result, the hospital invested in an optical fiber backbone, began to subject departmental systems to open systems requirements, and pressed the vendor of the main hospital-wide clinical information system to make its data more accessible.

Reference

- [1]. J. Ousterhout. Tcl and the Tk Tool kit. Addison Wesley, Reading, MA, 1994.